REMARKS

Status of Claims

Claims 1-33 are currently pending with claims 1, 6, 9, 14, 19, 24, and 29 being independent.

Office Action

In the December 21, 2005, Office Action, the Examiner rejected claims 1-3, 6, 9-11, 14-16, 19-21, 24-26, and 29-31 under 35 USC 102(e) as being anticipated by Goldberg (U.S. Patent No. 6,496,833), rejected claims 4, 8, 12, 17, 22, 27, and 32 under 35 USC 103(a) as being unpatentable over Goldberg in view of Steele (U.S. Patent Application Publication No. 2001/0056420), and rejected claims 5, 8, 13, 18, 23, 28, and 33 under 35 USC 103(a) as being unpatentable over Goldberg in view of Agesen (U.S. Patent No. 6,711,672). Applicant respectfully submits that the cited art does not disclose each and every element as claimed.

The Examiner generally contends that Goldberg, columns 11-12, discloses receiving a query, interpreting the query by associating a declarative language function with the query, converting the interpreted query to an imperative language statement, and executing the imperative language statement (Office Action, page 3). Applicant respectfully submits that Goldberg does not disclose or suggest these or other claimed features, as Goldberg is limited to converting a query from a first query language (SQL) to a second query language (CORBA) utilizing source code.

For example, Goldberg discloses receiving a query string (SQL command) from a user (col. 11, line 21) and using a code generator to generate source code to implement a query object (a CORBA object) (col. 11, lines 26-33). Thus, Goldberg only discloses converting one query (a SQL command) into another (a CORBA object) and does not disclose converting a query into an imperative language statement (a C function for instance). The "source code" cited by the Examiner and disclosed by Goldberg is merely "code which is compatible with the underlying database and written in the implementation language which is used to implement the CORBA server" (col 12, lines 12-15). Goldberg provides no disclosure or suggestion that the

"implementation language" is anything but a query language (CORBA). As such, Goldberg provides no disclosure or suggestion of converting queries into an imperative language statement as Goldberg does not disclose or suggest any imperative language.

Further, Goldberg is directed at the problem of database compatibility (col. 2, lines 47-63). The portions of Goldberg cited by the Examiner, specifically columns 11 and 12, convert one query language (SQL) to another (CORBA) to ensure database compatibility (col 12, lines 9-15). As such, Goldberg is not concerned with converting query languages to imperative languages. In contrast, the claimed invention is not so limited as it converts a query in a query language (SQL) to an imperative language statement (a C++ statement) to enable the query to be executed by a device lacking the computing power to process the query language.

Additionally, claim 1 recites more than converting a query to an imperative language statement as it includes the feature of "interpreting the queries by associating at least one declarative language function with the query terms." Independent claims 9, 14, 19, 24, and 29 recite similar features. In contrast, Goldberg is entirely silent regarding declarative languages. For instance, assuming *arguendo* that Goldberg suggests converting a query statement to an imperative language statement, Goldberg would do so by direct conversion (SQL query string -> CORBA object) utilizing the source code generator 604 (col 11, lines 30-36).

The Examiner cites col. 11, lines 44-57 for the premise that Goldberg discloses associating a declarative language function with query terms. However, this cited portion of Goldberg merely discloses creating a query object class that translates parameter notations between a notation entered by a user and the notation actually used by the database (col. 11, lines 49-51). For instance, Goldberg provides the example of a QueryInfo class which can translate an SQL colon (:) into a JDBC database format (?). Such translation between query formats is not the same as associating a declarative language function with a query. Further, Goldberg's direct conversion of a query string to a query object (SQL string -> CORBA object) teaches away from the use of an intermediate language (such as a declarative language). Thus, Goldberg does not disclose or suggest all features recited in the claims.

The inclusion of Steele and Agesen in the various rejections under 35 USC 103 does not cure the deficiencies of Goldberg. Specifically, Steele and/or Agesen do not disclose or suggest receiving query statements, interpreting, converting, or representing the query statements with declarative language functions, or converting declarative language functions to an imperative language. Further, the Examiner does not contend that Steele and/or Agesen disclose such features as neither Steele nor Agesen disclose queries of any sort.

The dependent claims recite further claim elements neither taught nor suggested by the art of record. For example, claims 2, 15, and 25 recite "converting the query language to an intermediate tree representation corresponding to the at least one declarative language function associated with the plurality of query terms, and thereafter converting the query to at least one data structure that is interpreted by an imperative language interpreter core to perform the queries," which is neither taught nor suggested by any reference of record. Claim 10, 20, and 30 recite a similar claim element and are similarly patentable for at lease these reasons. The Examiner's contention that Goldberg (columns 11-12) discloses this feature is incorrect as the cited portion of Goldberg merely discloses converting a first query element (a SQL string) into a second query object (a CORBA object) and not an intermediate tree representation corresponding to a declarative language function as recited in the claims.

Conclusion

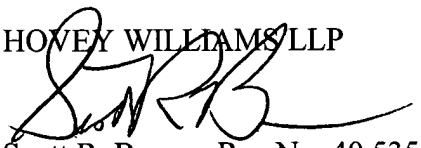
The Examiner's cited references, Goldberg, Steel, and Agesen, alone or in combination, fail to disclose or suggest all features recited in the claims. Specifically, Goldberg merely provides a method of converting one query language statement (a SQL string) to another (a CORBA object) and does not interpret, convert, or otherwise represent query statements utilizing declarative and imperative languages. Accordingly, Applicants respectfully request that a timely Notice of Allowance be issued in this case.

Should the Examiner have any questions, please contact the undersigned at (800) 445-3460. While the undersigned does not believe any additional fees are due in connection with this

application, the Commissioner is hereby authorized to charge any additional fees associated with this communication or credit any overpayment to Deposit Account No. 09-0460.

Respectfully submitted,

HOVEY WILLIAMS LLP

By: 

Scott R. Brown, Reg No. 40,535
HOVEY WILLIAMS LLP
2405 Grand Boulevard, Suite 400
Kansas City, Missouri 64108
(816) 474-9050

ATTORNEYS FOR APPLICANTS